作业 06　MIPS 处理器设计

1、

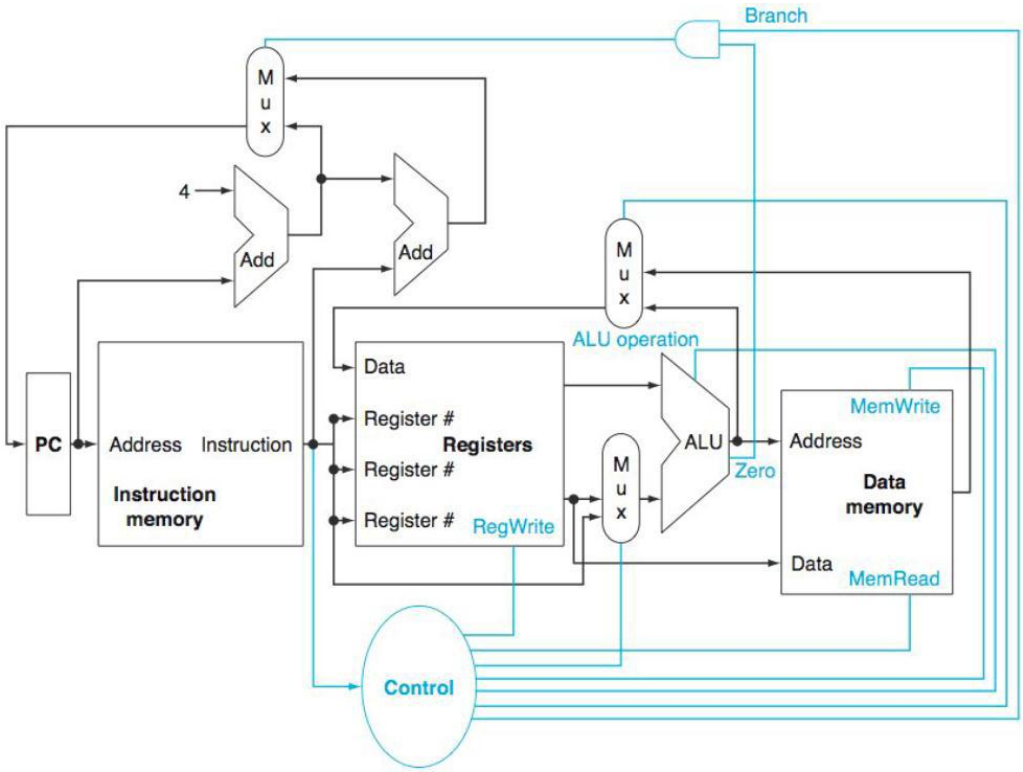

图 1

在基本的单周期 MIPS 实现中，不同的指令使用不同的硬件单元。

根据如下指令回答下列 3 个问题。

	指令	解释
a.	add Rd,Rs,Rt	Reg[Rd]=Reg[Rs]+Reg[Rt]
b.	lw Rt,Offs(Rs)	Reg[Rt]=Mem[Reg[Rs]+Offs]

(1)对上述指令而言，图 1 中的控制单元要产生哪些控制信号？

(2)对上述指令而言，要用到哪些功能单元？

(3)哪些功能单元会产生输出，但输出不会被以上指令用到？对以上指令而言，哪些功能单元不产生任何输出？

不同单元有不同的延迟时间。在图 1 中有七种主要单元。对一条指令而言，关键路径(产生最长延迟的那条路径)上各个单元的延迟时间决定了该指令的最小延迟。假设个单元的延迟时间如下表所示，回答下列 3 个问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	100ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	65ps

(4)对一条 MIPS 的与指令(AND)而言，关键路径是什么？

(5)对一条 MIPS 的装载指令(LW)而言，关键路径是什么？

(6)对一条 MIPS 的相等则分支指令(BEQ)而言，关键路径是什么？

答：

（1）

a：RegWrite = 1；其余为 0

b：RegWrite = MemRead = ALUSrc（Imm） = MemReg = 1；其余为 0

（2）

a：除 DM 和分支加法器

b：除分支加法器

（3）

产生输出，但不会被用上

a：分支加法器

b：分支加法器，寄存器堆的 Rt 输出

不产生任何输出

a：DM

b：无

（4）指令存储器 - 寄存器堆 - 多选器 - ALU - 多选器 - 寄存器堆

（5）指令存储器 - 寄存器堆 - 多选器 - ALU - 数据存储器 - 多选器 - 寄存器堆

（6）指令存储器 - 寄存器堆 - 多选器 - ALU - 多选器

2、图 1 中基本的单周期 MIPS 实现仅能实现某些指令。可以在这个指令集中加入新的指令，但决定是否加入取决于给处理器的数据通路和数据通路增加的复杂度。对于下表中的新指令而言，试回答下列 3 个问题。

	指令	解释
a.	add3 Rd,Rs,Rt,Rx	Reg[Rd]=Reg[Rs]+Reg[Rt]+Reg[Rx]
b.	sll Rt,Rd,Shift	Reg[Rd]=Reg[Rt]<<Shift(左移)

(1)对上述指令而言，哪些已有的单元还可以被使用？

(2)对上述指令而言，还需要增加哪些功能单元？

(3)为了支持这些指令，需要在控制单元增加哪些信号？

当设计者考虑改进处理器数据通路时，往往要考虑性能与成本的折中。假设我们从图 1 的数据通路出发，其中指令存储器(Instruction Memory)、加法器(Add)、多选器(Mux)、ALU、

寄存器堆(Registers)、数据寄存器(Data Memory)和控制单元(Control)的延迟分别为 400ps、100ps、30ps、120ps、200ps、350ps 和 100ps，相应的成本分别为 1000、30、10、100、200、2000 和 500。试根据表中的改进分别回答下列问题。

	改进	延迟	成本	优势
a.	更快的加法器	加法单元-20ps	每个加法单元+20	把已有的加法器用更快的加法器替代
b.	更大的寄存器堆	寄存器堆+50ps	寄存器堆+150	需要更少的load和store指令。这将导致指令数减少5%

(4)改进前后的时钟周期分别是多少？

(5)改进后将获得多大的加速比？

(6)比较改进前后的性能/价格比，进行这样的改进是否有意义？

答：

（1）

除 DM 和分支加法器之外的其他单元都可以被使用

（2）

a: 寄存器堆需要新加一个 Rx 的读入和 R[Rx]的输出，另外需要在后边新加一个 ALU 用于运算以及多路选择器判定是否传到后续单元

b: 在 ALU 中增加左移操作

（3）

a: 只需要对于 ALUop 新增 3 个相加的控制信号

b: ALUop 新增左移信号

（4）

改进前（如 lw 指令）：400 + 200 + 30 + 120 + 350 + 30 + 200 = 1330ps

改进后：

a: 由于最长数据通路不包含加法器，故无变化

b: 由于寄存器堆+50ps，故总时间为：1430ps

（5）

a: 1

b: 1 * 1330 / （1430 * 0.95） = 0.979

（6）

价格：

改进前：1000 + 3 * 10 + 100 + 200 + 2000 + 500 + 2 * 30 = 3890

改进后：

a: 3930 - 价格比：3930 / 3890 = 1.010 - 性能 / 价格比 = 1 / 1.010 = 0.990

b: 4040 - 价格比：4040 / 3890 = 1.039 - 性能 / 价格比 = 0.979 / 1.026 = 0.942

故可知，两个方案改进都无意义

3、

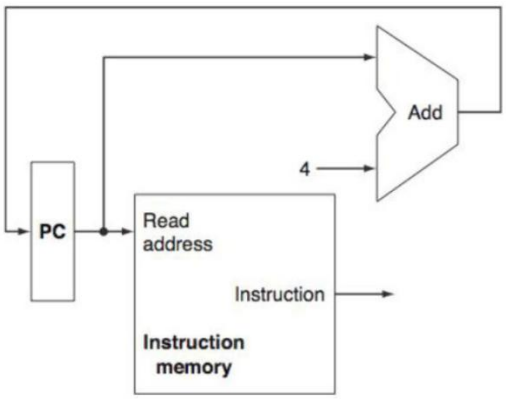

图 2

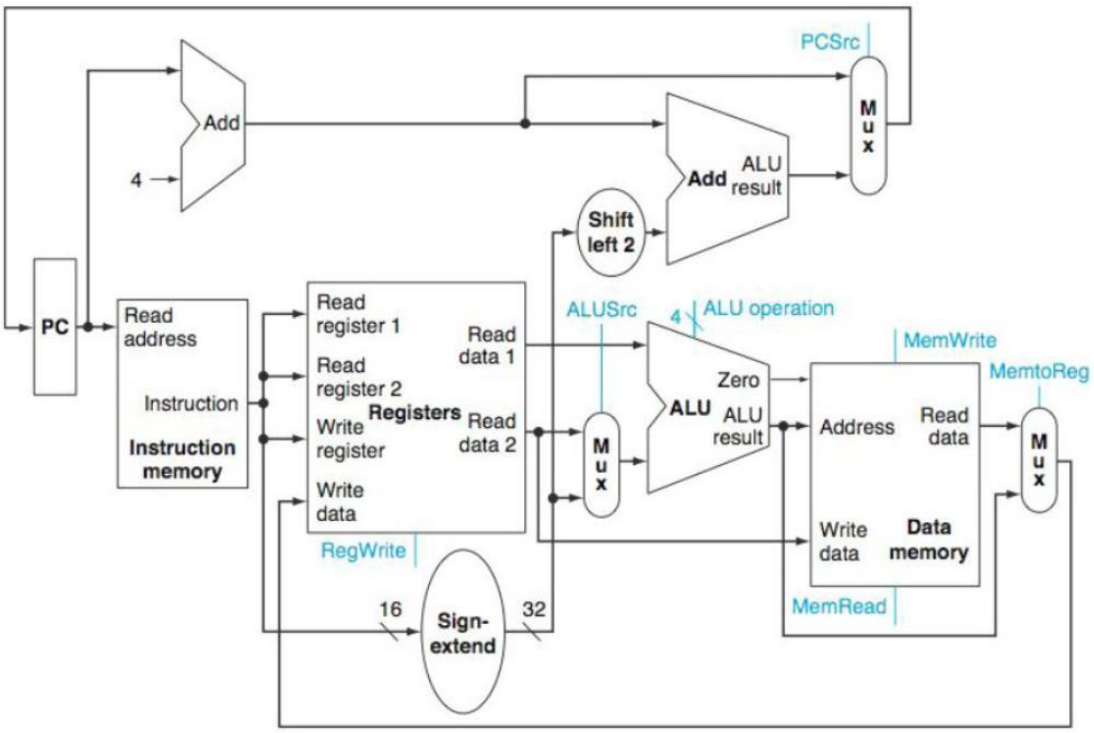

图 3

下表给出了实现处理器数据通路的逻辑单元延迟。试根据下表的两种情况分别回答下列问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	30ps	2ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

(1)如果处理器只需做连续取指这一件事(见图 2)，那么时钟周期是多少？

(2)考虑一个与图 3 类似的数据通路，但是假设处理器只需处理无条件相对跳转指令，那么时钟周期是多少？

(3)同样考虑一个与图 3 类似的数据通路，但这次假设只需处理有条件相对跳转指令，

那么时钟周期是多少？(请注意图 3 中 ALU 的零输出端不是与数据存储器连接，该输出与选择 PC 值来源的多选器的控制有关)（提示：图 3 中靠右侧的加法器延迟应当按照 ALU 来计算）

根据下表的两种数据通路的逻辑单元，分别回答下列问题。

	单元
a.	执行加4的加法器(对PC)
b.	数据存储器

(4)哪些类型的指令需要该单元？

(5)对哪些类型的指令而言，该单元位于关键路径上？

(6)假设仅需支持 beq 指令和 add 指令，讨论该单元的延迟变化对处理器时钟周期的影响。假设其他单元的延迟不变。

答：

（1）

a: 400ps

b: 500ps

（2）

a: 400 + 30 + 2 + 120 + 30 = 582ps

b: 500 + 90 + 20 + 180 + 100 = 890ps

（3）

a: 400 + 200 + 30 + 120 + 30 = 780ps

b: 500 + 220 + 100 + 180 + 100 = 1100ps

（4）

a: 所有指令

b: 需要访问内存的指令，如 lw、sw

（5）

a: 不在所有指令的关键路径上

b: 需要访问内存的指令

（6）

beq 指令周期：400 + 200 + 30 + 120 + 30 = 780ps

add 指令周期：400 + 200 + 30 + 120 + 30 + 200 = 980ps

a: 由于涉及 PC+4 加法器路径的时间远小于 beq 指令时间，且不影响符号扩展路径的时间，故不会造成影响

b: 因为 beq 指令和 add 指令都不涉及 DM 单元，故不会有影响

4、本题讨论数据通路中不同的单元延迟对整个数据通路时钟周期的影响，以及指令如何利用不同的数据通路单元。根据下面的两种延迟情况，分别回答下列问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	30ps	0ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

(1)如果仅需支持 ALU 类指令(如 add、and 等)，处理器的时钟周期是多少？

(2)如果仅需支持 lw 类指令，时钟周期是多少？

(3)如果必须支持 add、beq、lw 和 sw 指令，时钟周期是多少？

假设各类型指令所占比例如下表所示，试根据下表的两种情况分别回答下列问题。

	add	**addi**	**not**	**beq**	**lw**	**sw**
a.	30%	15%	5%	20%	20%	10%
b.	25%	5%	5%	15%	35%	15%

(4)数据存储器平均用了多少时钟周期？

(5)符号扩展电路的输入平均用了多少时钟周期？在未用到该输入的其他时间，符号扩展电路在做什么？

(6)如果可以将数据通路上某个单元的延迟减少 10%，应该减少哪个单元的延迟？改进后整个处理器的加速比是多少？

答：

（1）

a: 400 + 200 + 30 + 120 + 30 + 200 = 980ps

b: 500 + 220 + 100 + 180 + 100 + 220 = 1320ps

（2）

a: 400 + 200 + 30 + 120 + 350 + 30 + 200 = 1330ps

b: 500 + 220 + 100 + 180 + 1000 + 100 + 220 = 2320ps

（3）

由于 lw 指令的时钟周期为最长，所以在单周期而言，以 lw 为固定时钟周期

a: 1320ps

b: 2320ps

（4）

仅有 lw 和 sw 指令会使用一次 DM 单元

a: 20% + 10% = 30%时钟周期

b: 35% + 15% = 50%时钟周期

（5）

addi、beq、lw、sw 会用到符号扩展单元

a: 15% + 20% + 20% + 10% = 65%

b: 5% + 15% + 35% + 15% = 70%

在为用到该输入的其他时间，仍然会进行符号扩展，但不会被并且进行 PC 加法，但不会被多选器选择

（6）
由于处理器的时钟周期收到 lw 指令的影响，故将其使用的单元进行优化是最优的
对于 a 考虑优化指令存储器（400ps）或者寄存器堆（200ps * 2）
对于 b 考虑优化数据存储器（1000ps）

a: 改进后时钟周期：1290ps - 加速比：1330 / 1290 = 1.031

b: 改进后的时钟周期：2220px - 加速比：2320 / 2220 = 1.045

5、在制造硅芯片时，材料的缺陷和制造错误会导致电路失效。一个非常普遍的问题是一根线上的信号会对相邻线上的信号产生影响，这被称为串扰。有一类串扰问题是这样的，某些线上的信号为常值(如电源线)，该线附近的线也被固定为 0(stuck-at-0)或 1(stuck-at-1)。试根据下表的两种缺陷(信号来自图 4)分别回答下列问题。

	有问题的信号
a.	指令存储器，输出信号第7位
b.	控制单元，输出信号MemtoReg

(1)设这样测试处理器的缺陷：先给 PC、寄存器堆、数据和指令存储器中设置一些值(可以自己选择)，执行一条指令，然后读出 PC、寄存器堆和存储器中的值；最后检查这些值以判断处理器中是否存在缺陷。你能设计这样一个方案检查该信号上是否有固定为 0 缺陷吗？

(2)条件同第①问，但是这次检查固定为 1 缺陷。你能只设计一个测试方案同时检查固定为 0 缺陷和固定为 1 缺陷吗？如果可以，请解释如何实现；如果不能，请说明理由。

(3)如果我们知道一个处理器在该信号上有一个固定为 1 缺陷，它还能用吗？为了使这个处理器仍然可用，我们必须将原来能在正常 MIPS 处理器上运行的程序做一些变换，使之可以在这个处理器上运行。假设指令存储器和数据存储器都很大，足够容纳变换后的程序。提示：将因为该缺陷不能用的指令替换为一系列能用的指令，这一系列指令与原指令功能相同。

根据下表的缺陷分别回答下列问题。

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

(4)条件同第①问，这次检测控制信号 MemRead 是否存在上表中的缺陷？

(5)条件同第①问，这次检测控制信号 Jump 是否存在上表中的缺陷？

(6)使用第①问中描述的测试方案，可以一次对几个不同的信号进行测试，但一般来说不可能同时测试到所有信号。试着设计一系列方案对所有多选器输出的上表中的缺陷进行测试(五个多选器输出的每一位都要测试到)。尽量使用较少的测试方案。

答：

（1）

a: 由于是机器码的第七位，故可以使用 I 型指令 addi $t1 $0 128，如果有故障，则$t1 输出固定是 0，否则应该是 128

b: 当 MemtoReg 为 0 时，无法从 DM 中读取数据，故可以用 lw $t1, 0($zero)，取内存中已存好的值，如果有故障，则$t1 无法得到预定值

（2）

a: 同（1）理，addi $t1, $0, 0，如果有故障 1 故障，$t1 为 128；再使用 addi $t1, 0, 128，如果有 0 故障，则$t1 为 0

b: add $t1, $0, $t1，如果$t1 不是$0 + $1 的值，则存在 1 缺陷；lw $t1, 0($0)，如果$t1 的值不是内存中对应字段的值则存在 0 缺陷

（3）

a: 对于 I 型指令，我们可以将立即数第 7 位进行特殊处理，如果为 0，则可将立即数先先减去 128，后续进行此指令操作；对于 R 型指令，我们判断 shamt 编号是否存在二进制第一位为 0 的情况，否则，可以先对其进行-2 操作；对于 J 型指令，如果可以得到 pc 相对跳转的地址，则可以进行与 I 型同理的操作

b: 无法通过变换指令来修复，因为所有的 R 型指令都无法通过 ALU 将值写入寄存器

（4）

a: 不会对处理器有影响

b: 因为只有 R 型指令的 31 - 26 位全是 0，但对于内存访问的指令前六位都不全是 0，故不会对处理器有任何影响

（5）

a: 对于 addi $1, $0, 1 如果操作之后 PC = 0x00020008 则有缺陷，如果为 0x00000004 则无缺陷

b: 由于仅有 R 型指令前 6 位全为 0，而此时用不到 Jump 控制信号，故无法进行检测

（6）

	a	b
RegDst	lw $1, 8($0)， rd 端口会读到 00010，如果 RegDst 选 1 的话，则存入$2 中，无缺陷则存$1 中	add $2, $1, $1 如果$2 正常变化为$1 的两倍则无缺陷，否则如果$1 变为原来的两倍则有缺陷
ALUSrc	add $1, $0, $0 如果$1 为 0 则正常，否则由于立即数 0x0820，如果$1 为 0x00000820，则有缺陷	无法检验
MemtoReg	由前问可知，无法检验	无法检验
Branch	对于 sub $1, $0, $0 如果操作之后 PC 不为 0x00000004，则有缺陷	无法检验

Jump	由前问可知 addi $1, $0, 1 如果操作之后 PC = 0x00020008 则有缺陷，如果为 0x00000004 则无缺陷	无法检验

6、本题讨论特定指令在单周期数据通路中的操作。根据下表中的 MIPS 指令分别回答下列问题(参考 MIPS 汇编作业中的附录)。

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

(1)该指令字的值是多少？

(2)提供给寄存器堆“Read register 1”端口的寄存器号是多少？该寄存器真的被读了吗？对于“Read register 2”呢？

(3)提供给寄存器堆“Write register”端口的寄存器号是多少？该寄存器真的被写了吗？

不同的指令需要设置数据通路上不同的控制信号。根据下表的两种控制信号情况分别回答下列问题(参考图 4)。

	控制信号1	控制信号2
a.	RegDst	MemRead
b.	RegWrite	MemRead

(4)对于前面的指令而言，这两个控制信号的值应该是多少？

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

(5)对图 4 中的数据通路而言，写出控制单元中实现这两个信号的逻辑表达式。假设我们仅需支持 lw、sw、beq、add 和 j(jump)指令。

答：

（1）

a: 100011 00110 00001 0000000000101000

0x8cc10028

b: 000101 00001 00010 1111111111111111

0x1422ffff

（2）

a: $6，是的

b: $1，是的

（3）

a: $1，是的

b: $2，因为 bne 是 I 型指令，write register 由 rt 决定，故次数为$2，但由于没有写的指令，故没有被写

（4）

a: 0 1

b: 0 0

（5）

a:

RegDst = !Op0!Op1!Op2!Op3!Op4!Op5

MemRead = Op0Op1!Op2!Op3!Op4Op5

b:

RegWrite = !Op0!Op1!Op2!Op3!Op4!Op5 + Op0Op1!Op2!Op3!Op4Op5

MemRead = Op0Op1!Op2!Op3!Op4Op5

主控单元真值表

Input or output	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

ALUOp指明ALU的运算类型
- 00，访存指令所需加法
- 01，beq指令所需减法
- 10，R型指令功能码决定

提供给ALU控制单元

51

7、本习题讨论流水线对处理器时钟周期的影响。表中给出了数据通路中不同阶段延迟的两种情况，试根据这两种情况分别回答下列问题。

	IF	ID	EX	MEM	WB
a.	300ps	400ps	350ps	500ps	100ps
b.	200ps	150ps	120ps	190ps	140ps

(1)流水线处理器与非流水线处理器的时钟周期分别是什么？

(2)lw 指令在流水线处理器和非流水线处理器中的总延迟分别是多少？

(3)如果可以将原流水线数据通路的一级划分为两级，每级的延迟是原级的一半，会选择哪一级进行划分？划分后处理器的时钟周期为多少？

假设处理器执行的指令比例下表两种情况所示，试根据每种情况分别回答下列问题。

	ALU	beq	lw	sw
a.	50%	25%	15%	10%
b.	30%	25%	30%	15%

(4) 假设没有阻塞和冒险，数据存储器的利用率是多少（占总周期数的百分比）？

(5) 假设没有阻塞和冒险，寄存器堆的写寄存器端口的利用率是多少？

答：

（1）

a:

流水线处理器：500ps

非流水线处理器：300 + 400 + 350 + 500 + 100 = 1650ps

b:

流水线处理器：200ps

非流水线处理器：200 + 150 + 120 + 190 + 140 = 800ps

（2）

a:

流水线：500 * 5 = 2500ps

非流水线：1650ps

b:

流水线：200 * 5 = 1000ps

非流水线：800ps

（3）

a: MEM，400ps

b: IF，190ps

（4）

a: 15% + 10% = 25%

b: 30% + 15% = 45%

（5）

a: 50% + 15% = 65%

b: 30% + 30% = 60%